AN IMPROVED FILE CARVER OF INTERTWINED JPEG IMAGES USING X_myKarve

NURUL AZMA ABDULLAH

A thesis submitted in
fulfilment of the requirement for the award of the
Doctor of Philosophy.

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia

JUNE 2014

ABSTRACT

File carving is a common technique for retrieving evidence data from computers that have been used for crime activities to assist crimes investigations especially in solving pornography cases where traditional data recovery fail. However, carving fragmented JPEG files are not easy to solve due to the complexity of determining the fragmentation point. In this research, X_myKarve's framework is introduced to address the fragmentation issues that occur in JPEG images. The framework consists of six steps namely, dataset acquisition and preparation, pre-processing, work instruction generation, image carving and reconstitution, image completeness validation and fragmentation handling. X_myKarve is extended using myKarve's framework by introducing a new technique, deletion by binary search to detect fragmentation point which is used to separate a file into several individual fragments. These fragments are then reassembled with the correct pairs which form a complete and correct image. X_myKarve is tested using various datasets namely DFRWS 2006, DFRWS 2007 and additional datasets which are prepared and designed to simulate a particular fragmentation problems addressed in this research. The result shows that X_myKarve is capable of carving 23.8% more than myKarve and 45.4% more than RevIt for DFRWS 2006 datasets where X_myKarve can carve intertwined fragmented JPEG images completely compared to myKarve and RevIt. X_myKarve is a good alternative for carving more fragmented JPEG files that are intertwined with each other.

ABSTRAK

Ukiran fail adalah satu teknik yang lazim digunakan bagi mendapatkan semula bukti dalam bentuk data digital dari komputer yang telah digunakan untuk aktiviti-aktiviti jenayah bagi membantu siasatan jenayah terutama dalam menyelesaikan kes-kes pornografi di mana pemulihan data secara tradisional gagal. Walaubagaimanapun, proses untuk mengukir fail-fail JPEG yang terputus adalah tidak mudah disebabkan kesukaran untuk menentukan titik pemutusan bagi fail-fail tersebut. Dalam kajian ini, rangka kerja bagi X_myKarve diperkenalkan dalam menangani masalah imej pemutusan fail yang berlaku dalam imej-imej JPEG. Rangka kerja ini terdiri daripada enam langkah iaitu, penyediaan set data, pra-pemprosesan, generasi arahan kerja, ukiran imej dan penyusunan semula, imej kesempurnaan pengesahan dan pengendalian pemecahan. X_myKarve dilanjutkan daripada rangka kerja bagi myKarve dengan memperkenalkan satu teknik baru, penghapusan dengan menggunakan carian binari untuk mengesan titik pemutusan yang digunakan untuk memisahkan satu fail yang diperolehi kepada beberapa serpihan. Serpihan-serpihan ini kemudian akan dicantum semula dengan pasangan yang betul bagi membentuk imej yang lengkap dan betul. X_myKarve kemudian ditentusahkan dengan menggunakan beberapa set data iaitu set data DFRWS 2006, DFRWS 2007 dan set data tambahan yang disediakan dan direka untuk simulasikan masalah pemutusan fail seperti yang ditangani oleh kajian ini. Hasil perbandingan menunjukkan X_myKarve mampu untuk mengukir imej-imej dalam set uji DFRWS 2006 lebih 23.8% berbanding myKarve dan lebih 45.4% berbanding Revit, di mana X_myKarve berjaya mengukir fail-fail imej JPEG yang terputus dengan lengkap berbanding myKarve dan RevIt. X_myKarve merupakan alternatif yang baik dalam mengukir lebih banyak imej-imej JPEG di dalam senario-senario yang melibatkan fail-fail imej JPEG yang terputus dan terkait di antara satu sama lain.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF SYMBOLS AND ABBREVIATIONS

ADB	-	Address Database
APP	-	Application segment
AWQ	-	Automated Work Queue
DAC	-	Define Arithmetic Coding Table
BDCT	-	Block-based Discrete Cosine Transform
BGC	-	Bifragment Gap Carving
BMP	-	Bitmap
CCITT	-	Consultative Committee on International Telegraphy and Telephony
COM	-	Comment
DCT	-	Discrete Cosine Transform
DF	-	Digital Forensics
DFRWS	-	Digital Forensics Research Conference
DHP	-	Define Hierarchical Progression
DHT	-	Define Huffman Table
DOC	-	Microsoft Words
DRI	-	Define Restart Interval
DQT	-	Define Quantization Table
EOF	-	End of File
EOI	-	End of Image
ERC	-	Expand Reference Component
Exif	-	Exchangeable image file format
FBI	-	Federal Bureau of Investigation
ISO	-	International Organization of Standard
ITU	-	International Telecommunication Union
JEIDA	-	Japan Electronics Industry Development

		Association
JFIF	-	The JPEG File Interchange Format
JPEG/JPG	-	Joint Photographic Experts Group
PDF	-	Portable Document Format
PPT	-	Microsoft Power Point
RST	-	Restartinterval Termination
SOF	-	Start of File/ Start of Frame
SOI	-	Start of Image
SOS	-	Start of Scan
UHP	-	Unique Hex Patterns
VJH	-	Validated JPEG Header
VJM	-	Validated JPEG Markers

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background of study

In the last few decades, computers' usage were limited where only the scientific community and governments applied them for scientific and security purposes. Nowadays, computers and Internet are widely accepted around the globe changing the way people using computer. Computers become personal and were used to assist us in general and personal tasks. However, these scenarios also give ample opportunity for criminals to introduce new ways of committing crimes (Sitaraman, 2006; Mitrakas, 2006; Browne, 1972). To make matter worse, crimes committed using computers or cybercrime are hard to be proven. Digital Forensics (DF) is a platform for recovery and investigation of material found in digital devices, usually in assisting investigation related to computer crime. In the earlier years, DF's role is limited yet with the increasing number of computer usages in our daily life, DF entered its Golden Age from 1999 to 2007. DF plays an important role not only in assisting in cracking cases against computers crimes like phishing or frauds but the most obvious is its role in retrieving evidence from computers that been used to commit crimes like money laundering and child exploitation. It becomes an important tool to reconstruct the evidence left by cyber perpetrator. There are two ways of recovering digital evidence, - traditional data recovery and file carving.

Traditional data recovery is a common method used to recover digital data where the metadata or file allocation table exists. On the other hand, file carving was introduced to assist in cases where the traditional data recovery method cannot be utilized. Carving is used to explain the process of extracting a raw image from

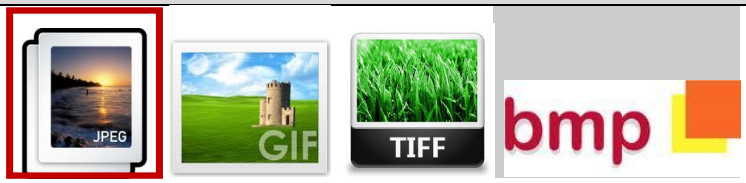
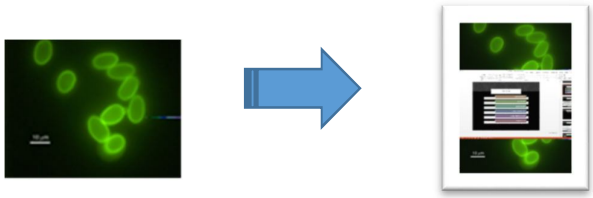
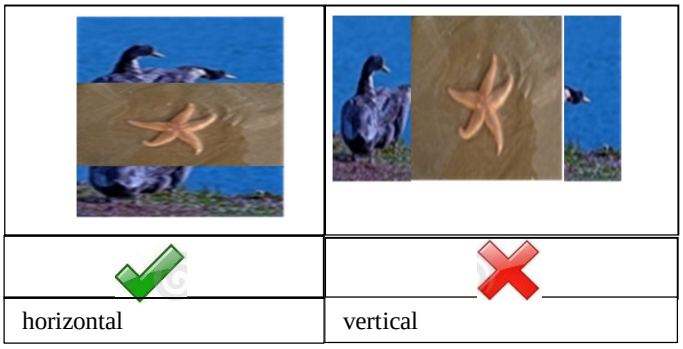
unstructured digital forensic images (Cohen, 2007b) based on the content rather than using a file system metadata for data recovery and computer forensics (Garfinkel, 2007). This is usually found in cases where file system metadata does not exist to provide direct recovery of data. Simply, file carving is a process of recovering files from a disk without knowing the file system (Veenman, 2007). There are many files that can be recovered from the target disk but most common files that are forensics interest are images. Hence, the main focus of this research is regarding preparing images as evidence data to assist in cybercrime investigation.

1.2 Motivation

Commonly, data evidence in cybercrime investigation involve image files. Therefore, most of the terminologies used in this research are concerning about images as illustrated in Table 1.1. In general, images can be stored in many formats such as Bitmap (BMP), TIFF, JPEG and etc. Out of these image formats, JPEG file is most commonly used in the internet because of its less structured and easily compressed features that can speed up internet transferring processes (Cohen, 2007b; Viraktamath & Attimarad, 2011; Li *et al.*, 2011). Nevertheless, in some cases, these images are fragmented when they are stored in the hard disk.

Fragmented files are hard to recover especially if they are intertwined among themselves which proves to be an obstacle to recovery. In a dataset, files can be contiguous or fragmented. However, according to Garfinkel (2007), less than 10% fragmentation occurred in a typical disk. Even though the percentage of fragmented files is relatively small but, these are usually the files that are of interest for forensics purposes. A file can be fragmented or split into two (bi-fragmentation) or more fragments, but multiple fragmentations (more than two fragments) require additional effort to be handled. In a disk, a file is stored by written the data from left to right of the disk. Therefore, when a file is fragmented, it will be fragmented horizontally and not vertically. Fragmentation can be linear or non-linear. A linear fragmentation is a condition where all fragments for a file present in a dataset in their original order while non-linear fragmentation is a condition where files are fragmented but some of the fragments present in different order not as in the original file.

Table 1.1: Research terminologies

Terminology	Image
Image Formats	
Fragmentation	 Complete file Fragmented file
Linear vs Non-linear fragmentation	 Linear fragmentation Non- linear fragmentation
Intertwined	 Simple linear intertwined Tightly linear intertwined
Fragmentation- Vertical vs horizontal	 horizontal vertical

There are two types of fragmentation for JPEG images that are being interest of this research - simple linear intertwined and tightly linear intertwined. Simple linear intertwined is when a JPEG file is split into two parts and fragmented with another JPEG image while tightly linear intertwined is when two JPEG files are split into two parts and fragmented with the other image's fragment as illustrated in Table 1.1.

In terms of carving tools, there are only a limited number of carving tools available today. Out of those, most tools concentrate on carving contiguous data file and they are not provided any validator which results in many false positive files (Garfinkel, 2007). In file carving discussion, a false positive file is a file contains some of the characteristics as the specific file types but does not relate to the file format (Metz & Mora, 2006). A signature in header and footer has been used to carve in straightforward carving. This is a simple technique and has been successfully proven to carve contiguous files with an assumption that the files clusters remain in order (Veenman, 2007). However, if files are fragmented, the files can be disconnected and becomes unordered, which causes the straightforward carving method to fail. Mohamad *et al.* (2009a) and Ying & Thing (2011) pointed out the importance of focusing on fragmentation problem especially within Define Huffman Table (DHT) area because any damage in DHT can cause image distortion or worse, corruption. Therefore, 2006 Carving Challenge organized by Carrier, Eoghan & Wietse (2006) initiate the efforts to encourage research on fragmented files carving by preparing data set containing some contiguous files while other files were fragmented. Although statistics presented in (Garfinkel, 2007) showed the fact that fragmentation in today's file system is relatively infrequent, the capability of carving fragmented files which is not extensively explored is important for computer forensic because the possibility of files that interest forensic investigation to be fragmented is relatively high. Mohamad *et al.* (2010b) proposed myKarve as a tool to carve contiguous and linearly fragmented images caused by other file formats which are called "garbages". This tool has successfully carved not only contiguous and linearly JPEG files but also thumbnails but limited to thumbnail with distinct marker from the parent. Even though myKarve works on fragmented JPEG files, but it only concentrates on JPEG files that is fragmented with other types of files such as Word, PDF and Excel. On the other hand, this research strives to study a way of carving JPEG files that are intertwined with another JPEG files with capability of carving thumbnails including those that have

similar marker with parent which is harder to carve if using standard header pattern matching.

There are two scenarios considered for fragmentation between JPEG files. The first scenario is where a fragmented JPEG file is intertwined with a complete JPEG file as shown in Figure 1.1. Consider two JPEG files, F1 and F2. In certain conditions as explained by Garfinkel (2007), a JPEG file is fragmented where it split into two parts. In Figure 1.1, F1 is linearly fragmented where it splits into two parts while the first part, $F1_1$ comes first before the second part, $F1_2$. This file is intertwined with a complete JPEG image, F2 where the second part of F1, $F1_2$ comes after F2 ends. This condition is called simple linear intertwined JPEG files.

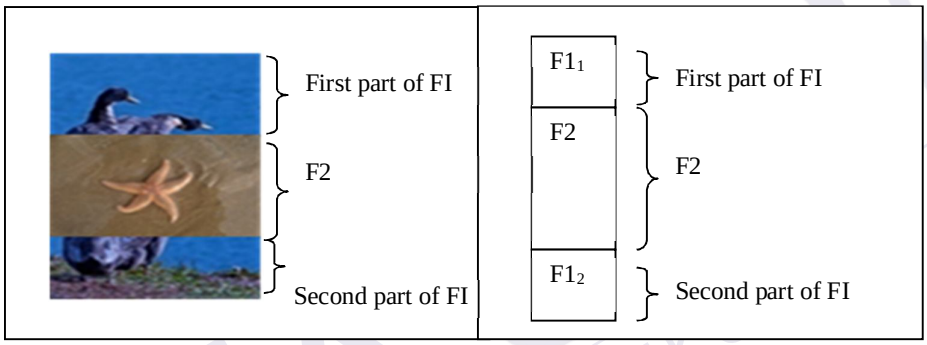


Figure 1.1: Simple linear intertwined JPEG files

In the second scenario, two linearly fragmented JPEG files are intertwined with each other. Figure 1.2 shows two JPEG files where the second part of F1, $F1_2$ comes after the first part of F2, $F2_1$ and $F1_2$ is located before the second part of F2, $F2_2$. In both scenarios, it is important to recognize the boundary of each file to distinguish between those two files. In addition, different approaches are required to carve the original JPEG files with or without thumbnail for both scenarios.

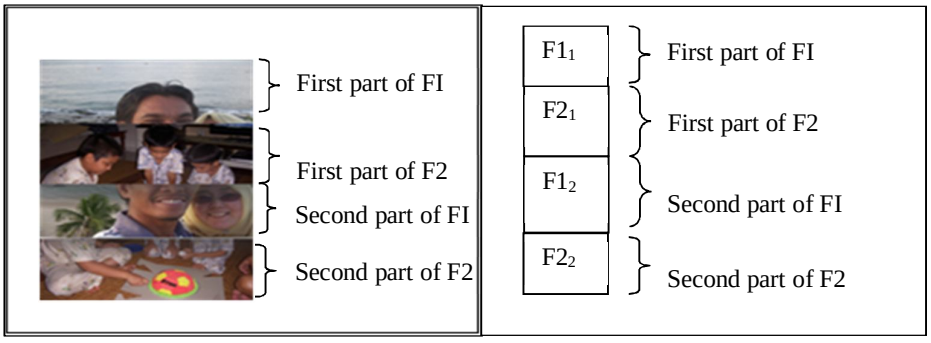


Figure 1.2: Tightly linear intertwined JPEG files

The term “parent JPEG image” or “original JPEG image” in this thesis is used to denote the original JPEG image that is neither embedded JPEG file nor thumbnail. Although myKarve is able to detect fragmented files but it only carves files that are fragmented by other types of files such as DOC, PPT and PDF. myKarve has not yet managed to address the scenario where JPEG files are fragmented with each other. Fragmentation with same file format is more difficult to handle because of the headers are similar and indistinguishable. To compensate with these limitations, this research strives to address those two scenarios discussed earlier by extending myKarve framework with added capability to identify fragmentation point for the fragmented JPEG files and then try to recover them.

1.3 Aim and objectives of study

The aim of this study is to provide a technique to address fragmentation scenarios where a JPEG file was split into two fragments and then intertwined with either a complete JPEG file or another bi-fragmented JPEG file.

The objectives of this research are:

(i) to propose a method by extending myKarve’s framework for handling and reassembling linearly simple and tightly intertwined JPEG files.
(ii) to develop an improved carving tool by implementing the proposed method.
(iii) to evaluate the performance of the proposed method in term of the success recovery number of intertwined JPEG files on DFRWS 2006 and DFRWS 2007.

1.4 Scope of study

This research focuses on extending myKarve’s framework for carving linearly simple and tightly intertwined JPEG files. This research does not include non-linear scenarios or cases where JPEG file were fragmented into more than two fragments or file with missing fragments. The proposed approach concentrates on getting higher successful carving rate of intertwined JPEG files. In this research, only a complete baseline JPEG

is considered because it is recommended to ensure maximum compatibility for file interchange. The research deals only with JPEG files without thumbnail and JPEG files with one thumbnail or two thumbnails. The proposed approach will also address the fragmentation scenario where the fragmentation point is located after SOS (Start of Scan) marker of a non-thumbnail image.

1.5 Organization of Thesis

The rest of the chapters in this thesis are organized as follows: Chapter 2 discusses in general of two fields in Digital Forensics, - the traditional data recovery and file carving, the JPEG standards, file fragmentation and various existing carving tools. Chapter 3 discusses about X_myKarve framework and algorithms, datasets preparation, predefined JPEG's scenarios patterns, carving and fragmentation handling processes. Chapter 4 discusses important processes in X_myKarve's implementation, carving experimentations and fragmentation handling. Chapter 5 discusses the results of the carving process and compare the developed method with other tools. Finally, Chapter 6 concludes the research and provide suggestions for future works.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

There are a number of definitions given on Digital Forensics (DF) / Cyber Forensics. Povar & Bhadran (2011) defined Cyber Forensics as a process of acquisition, authentication, analysis and documentation of evidence extracted from and/or contained in a computer system while Palmer (2001) defined Digital Forensics Science as "The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations".

DF has been introduced roughly within the last forty years. Garfinkel (2010) stated that it started with the need for data recovery. Among earlier researchers discussing on data recovery was Wood *et al.* (1987). The authors share a story of two local data recovery experts working 70 hours together to recover the only copy of highly fragmented database file that was carelessly deleted. However, the need for DF was limited due to the nature of computing environment at that time. According to Garfinkel (2010), in the last forty years, the main concerns are due to hardware, software and application diversity and various poorly documented data file formats. In this period, the computing environment was more to time sharing and centralized usage and there was no formal process, tools and training to support the environment. Forensics tasks were mainly handled by computer professionals who worked with law enforcement on ad hoc, case-by-case basis. Most of the time, according to Garfinkel

(2010), evidence left on time sharing systems could be recovered without the use of recovery tools. Furthermore, the disks were small which forces the perpetrators to make extensive printouts which limited the need for analysis of digital media. This is supported by a report in CVJCTS (2004) where only three cases were being examined in "Magnetic Media Program" initiated by the Federal Bureau of Investigation (FBI). Although computer hacking is a big concern to many organizations, but according to Computer Fraud and Abuse Act of 1984, during those years, computer hacking was not even a crime which limits the need of forensics analysis. Only the next pace the Golden Age of DF starts.

The golden Age of DF occurs from 1999 to 2007. This is when DF starts to emerge in an effort to mitigate the rates of cyber-crime. It is being used as a tool to look into the past through the recovery of residual data that was thought to have been deleted through the recovery of email and instant message. Two of the important fields in DF are data recovery and file carving. Data recovery is a process of recovering files using the file system metadata while file carving recovers files based on their content without using the file system metadata that point to the content.

2.1.1 Traditional data recovery

Pal & Memon (2009) discussed traditional data recovery that depends on file system structure to recover data that has been deleted. This is possible because of the nature of most file systems that are not doing anything to the physical location during file deletion; instead they simply mark the location as "unallocated" which indicate that it is available for storing data. The deleted file's information such as the information linking the clusters may still be present. In this situation, traditional data recovery can simply use the file system structures to recover the deleted file. However, when the file system metadata is not available or corrupted, then the traditional data recovery cannot be utilized. Here is where file carving's role is important to recover data in such situation.

2.1.2 File carving

Povar & Bhadran (2011) defined carving as a process of extracting data or file out of undifferentiated blocks or raw data while file carving as a process of identifying and recovering files based on analysis of file formats. Hence, file carving unlike traditional data recovery was introduced to recover data from a corrupted dataset where the file system metadata is not present (Garfinkel, 2007; Courrejou & Garfinkel, 2011; Zha & Sahni, 2010; Richard III *et al.* 2007; Ying & Thing, 2011). In other words, file carving is a process of recovering files based on their content without using the file system metadata that points to the content (Garfinkel, 2007; Pal *et al.*, 2008; Cohen, 2007b). It is important because traditional data recovery techniques are incapable of recovering any file without the file system metadata. Cohen (2007a) also pointed the importance of carving images with damaged or incomplete file system, for example, PhotoRec a popular carving tool introduced by Grenier (2007). PhotoRec can be used to recover photos from damaged memory cards. The taxonomy of file carving proposed by Simon Garfinkel and Joachim Metz is shown in Table 2.1.

File carving was initiated by The Defense Computer Forensics Lab (DCFL) that developed CarvThis, a carving program. Next, Agent Kriss Kendall, later join by Agent Jesse Kornblum introduced Foremost, which is an open source carving tool. Later, it was extended by Mikus (2005) by implementing a module with specific knowledge of Microsoft OLE. Richard and Roussev (2005) introduced Scalpel, an improvement of Foremost. The goal of Scalpel is to enhance performance and decrease memory usage. LibCarvPath and CarvFS are virtual file systems to provide zero-storage carving possibilities, developed by Dutch National Police Agency.

Year by year, with the increasing number of computers and other digital devices usages, file carving techniques also evolve drastically. The earliest carvers were simple Start of File/ End of File (SOF/EOF) carvers where these carvers simply searched the image for file headers and file footers. If the pairs are found, it extracts all the data in between them (Cohen, 2007b). At first, it was relatively satisfactory because file systems usually tried to keep files allocated consecutively to minimize fragmentation. However, there are some conditions where it is difficult to avoid fragmentation. Hence, the objective of later file carvers changes to take into account fragmentation where they will identify the indirect blocks that cause fragmentation and

then ignore these blocks when they encountered knowledge of the file system that is being used as applied in RevIt Smart (Metz & Mora, 2006).

Table 2.1: Carving techniques (source: Garfinkel (2012); Metx & Mora (2006))

Carving Technique	Description
Carving	General term for extracting data (files) out of undifferentiated blocks (raw data), like "carving" a sculpture out of soap stone.
Block-Based Carving	Any carving method (algorithm) that analyses the input on block-by-block basis to determine if a block is part of a possible output file. This method assumes that each block can only be part of a single file (or embedded file).
Statistical Carving	Any carving method (algorithm) that analyses the input on characteristic or statistic for example, entropy) to determine if the input is part of a possible output file.
Header/Footer Carving	A method for carving files out of raw data using a distinct header (start of file marker) and footer (end of file marker).
Header/Maximum (file) Size Carving	A method for carving files out of raw data using a distinct header (start of file marker) and a maximum (file) size. This approach works because many file formats (e.g. JPEG, MP3) do not care if additional junk is appended to the end of a valid file.
Header/Embedded Length Carving	A method for carving files out of raw data using a distinct header and a file length (size) which is embedded in the file format.
File Structure Based Carving	A method for carving files out of raw data using a certain level of knowledge of the internal structure of file types. Garfinkel called this approach "Semantic Carving" in his DFRWS2006 carving challenge submission, while Metz and Mora called the approach "Deep Carving."
Semantic Carving	A method for carving files based on a linguistic analysis of the file's content. For example, a semantic carver might conclude that six blocks of French in the middle of a long HTML file written in English is a fragment left from a previous allocated file, and not from the English-language HTML file.
Carving with Validation	A method for carving files out of raw data where the carved files are validated using a file type specific validator.
Fragment Recovery Carving	A carving method in which two or more fragments are reassembled to form the original file or object. Garfinkel previously called this approach "Split Carving."
Repackaging Carving	A carving method that modifies the extracted data by adding new headers, footers, or other information so that it can be viewed with standard utilities. For example, Garfinkel's ZIP Carver looks for individual components of a ZIP file and repackages them with a new Central Directory so that they can be opened with a standard unzip utility.

Cohen (2007) discussed the need to recover the targeted images fully not partially. This method can reduce the manual work of recovering other partial images. This is important due to the current typical hard disk size that contains hundreds of

thousands of files which makes manual examination impractical. This need was also stressed by DFRWS[1] that organized a competition aiming to improve the available carvers by concentrating in fully or semi-automated carving techniques. Then, the approach shifts to semantic carvers that use information about internal file structure to control the carving process. This was applied by Garfnkel (2007) with his object validator to detect corrupt files which reduce the incidence of false positives.

It is clear that file carving is important in both data recovery and computer forensics. There are steps in carving files of interest. According to Garfinkel (2007), there are general steps in file carving. First, files to be carved need to go through forensic imaging process to be recognized in the disk image. In this process, the entire drive's contents are imaged to a file. The images can be acquired with the use of software tools. Then, the files need to be processed to determine whether they are intact or not. Finally, the files need to be copied out of the disk image and presented to the examiner or analyst.

The most applied technique of file carving is by analyzing headers and footers of a file and try to merge all the blocks in between (Mohamad *et al.*, 2010b; Pal *et al.*, 2008). Three different algorithms in detecting header or footer for JPEG images are single-byte marker; 20 point references and dual-byte marker have been discussed in Mohamad & Mat Deris (2009b) and Mohamad *et al.* (2010a). Mohamad & Mat Deris (2009b) compared the performance of two algorithms for detecting JPEG JFIF header using FORIMAGE-JPEG. They proposed single-byte-marker algorithm against 20-point-reference algorithm. From the result, single-byte-marker algorithm outperforms 20-point-reference algorithm. Later, Mohamad *et al.* (2010a) proposed dual-byte-marker algorithm against single-byte-marker algorithm which was proved perform better than the earlier algorithm.

2.1.3 Earlier File Carving Tools

There are many file carvers that have been developed to date. Some earlier file carver tools have been listed in Table 2.2. There are few improvements that are important to enhance the efficiency of carving tools. According to Kloet (2007), there are two

[1] www.dfrws.org/2007/

conditions that need to be improved in developing carving tool which are “higher carving recall” and “higher carving precision”.

Table 2.2: File carving tools

Carver	Description
Simple Carver Suite[2]	Simple Carver Suite is a collection of unique tools designed for a number of purposes including data recovery, forensics computing and eDiscovery. The suite was originally designed for data recovery and has since expanded to include unique file decoding, file identification and file classification.
Foremost[3]	Foremost is a console program to recover files based on their headers, footers, and internal data structures.
Scalpel (Richard III & Roussev, 2005)	Scalpel is a fast file carver that reads a database of header and footer definitions and extracts matching files from a set of image files or raw device files. Scalpel is file system-independent and will carve files from FATx, NTFS, ext2/3, or raw partitions.
EnCase[4]	EnCase comes with some enScripts that do the carving.
CarvFs[5]	A virtual file system (fuse) implementation that can provide carving tools with the possibility to do recursive multi tool zero-storage carving (also called in-place carving). Patches and scripts for Scalpel and Foremost are also provided. Works on raw and encase images.
LibCarvPath[6]	A shared library that allows carving tools to use zero-storage carving on Carvfs virtual files.
RevIt (Metz & Mora, 2006)	RevIt (Revive It) is an experimental carving tool initially developed for the DFRWS 2006 carving challenge. It uses 'file structure based carving. Note that RevIt is currently a work in progress.
Adroit Photo Forensics[7]	Adroit Photo Forensics supports data carving of popular image formats. Also supports fragmented carving using” SmartCarving” and “GuidedCarving”.

“Higher carving recall” is about detecting as much useful information as possible and do not simply discards any interesting results. The goals as listed by Kloet (2007) are as follows:

- Support many file types to decrease the number of unsupported false negatives.
- Consider partial results and mark them as known false positives, since they might contain useful information.

[2] http://www.simplecarver.com/
[3] http://foremost.sourceforge.net/
[4] http://www.forensicswiki.org/wiki/EnCase
[5] http://www.forensicswiki.org/wiki/CarvFs
[6] http://ocfa.sourceforge.net/libcarvpath/
[7] http://digital-assembly.com/products/adroit-photo-forensics/features/smartcarving.html

- Carve corrupted files as known false positives and continue to carve a file even corruption detected to recover as much files as possible.

"Higher carving precision" is about carving known false positives with goals as follows (Kloet, 2007):

- Detect false positives and mark them as known false positives to reduce their negatives impacts.
- Better fragmentation handling than current tools where a fragmented file can be carved as a full file, instead as partial(s). This can increase the number of positives and decrease the number of false positives.

2.2 Overview of JPEG standard

Joint Photographic Experts Group (JPEG) format was formed by Consultative Committee on International Telegraphy and Telephony (CCITT) in 1986 inspired by an effort of International Organization of Standard (ISO) to find ways to use high resolution graphics and pictures in computers (Cohen, 2007a). CCITT is a permanent organization of the International Telecommunication Union (ITU) which is the United Nations Specialized Agency in the field of telecommunications. CCITT is the body which sets world telecommunications standard (CCITT, 1992). There are two types of JPEG, namely JPEG File Interchange Format (JFIF) and JPEG Exchangeable Image File Format (Exif) (Mohamad & Mat Deris, 2009c). JFIF is used for sharing in different applications and on the Internet while JPEG Exif is used for digital cameras (Alvarez, 2004; Mohamad & Mat Deris, 2009c). Exif was introduced by the Japan Electronics Industry Development Association (JEIDA) to encourage interoperability between imaging devices. Exif 2.1 was introduced in 1998 and the latest version, 2.4 in 2010. Exif has been accepted as the preferred image format for digital cameras universally, although not officially (Winzip Computing, 2008; Bettelli, 2006).[8]

JPEG introduced compression standard for both grayscale and color continuous-tone images. Compression is a technique where the quantity of multimedia data is being reduced without excessively reducing the quality in the data displayed. Compression allows faster process of storing and transitioning of multimedia data much faster than using the original data (Shahbahrami, 2011). Compression involves

[8] http://www.dpreview.com/learn/Glossary/Camera_System/*EXIF*_01.htm

two processes, encoding and decoding. Encoding is typically done in three steps - discrete cosine transform (DCT), quantization and entropy coding while for decoding process, the process are reversed (Fan & de Queiroz, 2003). Meanwhile, two classes of encoding and decoding processes, lossy and lossless have been introduced. Lossy is based on the DCT while lossless is not based on DCT (CCITT, 1992). While lossy allows substantial compression to be achieved, lossless is used to meet the needs of application requiring higher quality image. This method results in larger file size for lossless method, while lossy method produce file that will progressively lose its quality. Nevertheless, JPEG ISO standard is a lossy image compression. JPEG files consist of different functions such as color space conversion and entropy coding (Shahbahrami, 2011). Two entropy coding that are normally used in the entropy coding phase are Huffman and Arithmetic coding. Both lossless and lossy compression processes use these two encoding method. According to Shannon (2001), entropy is a measure of information density or compression state of a given unit of data. The entropy value will be lower if the file is less compressed. Hence, encrypted files have high entropy value compared to text files or Bitmap files. Comparison done by Shahbahrami (2011) showed that the compression ratio of arithmetic coding is better than Huffman coding, but its performance speed is slower and it is more difficult to implement. Hence, both coding can be applied depending on the user requirement. If high quality compression is required, arithmetic coding is recommended while for some applications that are time dependable, Huffman coding should be applied.

There are various ways to display lossy and lossless JPEG file on the screen. The ways of displaying both lossy and lossless JPEG image on the screen is called modes of operation.

2.2.1 JPEG modes of operation

There are four modes of operation as described in CCITT (1992) and Mohamad & Mat Deris (2009a) which are:

- Sequential DCT based mode (baseline)

  8 x 8 sample blocks are usually encoded block by block from left to right and block-row by block-row from top to bottom. This technique minimizes

coefficient storage requirements. Currently, most JPEG images are in this mode.

- Progressive DCT based mode

  It is encoded in the same order, but using multiple scan through the image. Hence, the transmission time is long and the image build up in multiple coarse-to-clear passes.

- Lossless mode

  The image produced using this mode of operation has higher quality compared to other methods because it guarantees exact recovery of every source image with no loss of quality.

- Hierarchical mode

  An image is encoded as a sequence of frames where these frames provide reference reconstructed components for prediction in subsequent frames. This method ensures lower-resolution versions may be accessed without need to be decompressed at its full resolution.

Although there are many compression modes supported by JFIF, Hamilton (1992) recommended the use of baseline or sequential DCT based mode to ensure maximum compatibility during the file interchange process. This is asserted by Wallace (1991), who claimed that sequential DCT based mode is the most widely implemented JPEG method to date. It is also appropriate to be used in a large number of applications. Nevertheless, every JPEG files are segmented by special two-byte codes called markers.

2.2.2 JPEG Markers

JPEG files are segmented by special two-byte codes called markers to identify various structural parts of the compressed data formats (CCIT, 1992). There are stand-alone markers, but most markers contain a related group of parameters. Some example of standalone markers are Start of Image (SOI), End of Image (EOI) and Restartinterval Termination (RST) while examples of markers containing a related group of parameters are Define Huffman Coding Table (DHT), Define Quantization Table (DQT), Start of Frame (SOF), Define Arithmetic Coding Table (DAC), Define

Hierarchical Progression (DHP), Expand Reference Component (ERC), Application segment (APP), reserved for JPEG extensions (JPG) and Comment (COM). A full list of these markers is shown in Table 2.3. Generally, a well formed JPEG file's structure is a collection of above markers. Next subsection will discuss on the detailed structure of JPEG file.

2.2.3 JPEG File Structure

There are two structures of JPEG file, JFIF and Exif. JFIF is common for application in the Internet while Exif is common for digital camera images. The detailed structure of two JPEG format was explained in the next paragraph.

2.2.3.1 JFIF

JFIF (JPEG File Interchange Format), the common format for JPEG data was introduced by Eric Hamilton in 1991 (Kornblum, 2008). It is a minimal file format that allows JPEG bitstreams to be exchanged in multiplatform environment and wide variety range of applications. According to work done by Hamilton (1992), the APP0 marker is inserted into JFIF structure as an additional requirement while maintaining the compatibility of JFIF with the standard JPEG format interchange. The detailed JFIF file header format is as in Table 2.4. A JPEG file that apply JFIF standard contains a header signature that starts with SOI followed by hexadecimal string 0xFF E0 XX XX 4A 46 49 46 00 and ends with EOI. The other format of JPEG is Exif as explained in the next subsection.

2.2.3.2 Exif

JPEG Exif standard was created to stipulate the method of recording image data in files which specifies the structure of image data files, tags used by this standard and, the definition and management of format versions. Exif can be recognized by SOI and "Exif\0" identifier. The basic structure of Exif files is as in Figure 2.1.

Table 2.3: Marker code assignments (Source: CCIT, 1992)

Code Assignment	Symbol	Description
Start of Frame markers, non-differential, Huffman coding		
xFFC0	SOF_0	Baseline DCT
xFFC1	SOF_1	Extended sequential DCT
xFFC2'	SOF_2	Progressive DCT
xFFC3	SOF_3	Lossless (sequential)
Start of Frame markers, differential, Huffman coding		
xFFC5	SOF_5	Differential sequential DCT
xFFC6	SOF_6	Differential progressive DCT
xFFC7	SOF_7	Differential lossless (sequential)
Start of Frame markers, non-differential, arithmetic coding		
xFFC8	JPG	Reserved for JPEG extensions
xFFC9	SOF_9	Extended sequential DCT
xFFCA	SOF_{10}	Progressive DCT
xFFCB	SOF_{11}	Lossless (sequential)
Start of Frame markers, differential, arithmetic coding		
xFFCD	SOF_{13}	Differential sequential DCT
xFFCE	SOF_{14}	Differential progressive DCT
xFFCF	SOF_{15}	Differential lossless(Sequential)
Huffman table specification		
xFFC4	DHT	Define Huffman table(s)
Arithmetic coding conditioning specification		
xFFCC	DAC	Define arithmetic coding conditioning (s)
Restart interval termination		
xFFD0 through xFFD7	RST_m*	Restart with modulo 8 count "m"
Other markers		
xFFD8	SOI*	Start of image
xFFD9	EOI*	End of image
xFFDA	SOS	Start of scan
xFFDB	DQT	Define Quantization table (s)
xFFDC	DNL	Define number of lines
xFFDD	DRI	Define restart interval
xFFDE	DHP	Define hierarchical progression
xFFDF	EXP	Expand reference component(s)
xFFEO through xFFEF	APP_n	Reserved for application segments
xFFF0 through xFFFD	JPG_n	Reserved for JPEG extensions
xFFFE	COM	Comment
Reserved markers		
X'FF01'	TEM*	For temporary private use in arithmetic coding
X'FF02' through X'FFBF'	RES	Reserved

Table 2.4 : JPEG JFIF segment header format (Hamilton, 1992; Mohamad & Mat Deris, 2009c)

Field	Size (byte)	Description
SOI marker	2	Always equal to FFD8
APP0 marker	2	Length of segment excluding APP0 marker
Length	5	Always equal to "*JFIF* " followed by 0x00 (or 0x$A 46 49 46 00) or a zero terminated string *JFIF*
Version	2	First byte is major creation (currently 0x01); second byte is minor version (currently 0x02) or version 1.02
Density units	1	Units for pixel density fields: • 0- No units, aspect ratio only specified • 1- Pixels per inch • 2- Pixel per centimetre
X density	2	Integer horizontal pixel density
Y density	2	Integer vertical pixel density
Thumbnail width *(tw)*	1	Horizontal size of embedded JPEG thumbnail in pixels.
Thumbnail height *(th)*	1	Vertical size of embedded JPEG thumbnail in pixels
Thumbnail data	3 x tw x th	Uncompressed 24 bit GB raster thumbnail

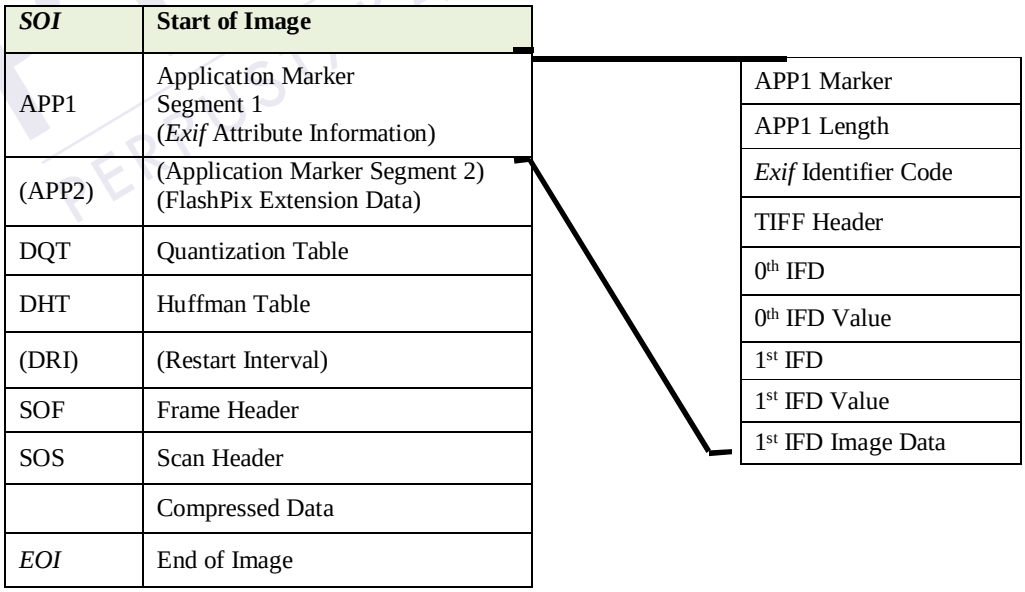


Figure 2.1: Basic structure of *Exif* files

A JPEG file that applies Exif standard contains a header signature that starts with SOI followed by hexadecimal string 0xFF E1 XX XX 45 78 69 66 00 and ends with EOI. The first eleven bytes of an Exif file header is shown in Table 2.5 (Mohamad, 2009c).

Table 2.5: First 11 bytes of a Exif file header with sample hexadecimal codes

SOI	***APP1***	**Length**	**Identifier**
0xFFD8 2 bytes	0xFFE1 2 bytes	0xXXXX (unknown two bytes) 2 bytes	0x 45 78 69 66 00 E x i f NULL 5 bytes

Both JFIF and Exif format allows for embedding thumbnail(s) into a JPEG file. The next section describes thumbnail and its role in assisting digital evidence preparation to prosecute cyber perpetrator.

2.2.4 Thumbnail(s) / Embedded JPEG images

A JPEG image with a complete SOI / EOI can be embedded into an original JPEG image to ease the process of recovering and organizing the original image. This file is known as thumbnail. Thumbnails are reduced size version of images that can be used to recover and organize a picture (Guo, 2011) while in the other hand, embedded JPEG files are the original JPEGs that are embedded to other types of files such as PPT, WORDS and EXCEL. Thumbnails are used to speed up image search or page load on the Internet and also being used in image organizing programs. Thumbnails are compatible on most modern operating systems or desktop environments such as Microsoft Windows, Mac OS X, KDE and GNOME. A JPEG image can contain none, single or two thumbnails (Mohamad *et al*., 2010b). Therefore, a JPEG image can have several SOI / EOI pairs (Merola, 2008). Mohamad *et al*. (2011) asserted the role of thumbnail to serve as a method of recognizing the corrupted images because of its small size giving it a better chance of full recovery without any corruption. A thumbnail carried similar features as the original. Hence, using thumbnail(s), crime investigators can identify which images or pictures that have the potential to be used as evidences against cyber perpetrator.

In addition, Guo (2011) proposed thumbnails as a method to recover JPEG image from fragmented data. In carving a JPEG image, it will help if the carver can distinguish between original images, thumbnails and embedded images. This information can help in reducing false fragmentation point detection caused by thumbnail that is mistaken as its original image due to its features that is similar to its parent. Hence, it is important to find any difference between original and thumbnail image. A file can be in two structures, non- fragmented and fragmented. Next section will further elaborate on these two types of file structures.

2.3 File structures

In file carving, there are two types of files structures that are normally found in a dataset which are non-fragmented file and fragmented file. A non-fragmented structure is when all data for the file is in the same or consecutive cluster while fragmented structure is when part of data for the file is split into different and non-adjacent cluster of the other part. The details of these two structures are discussed in the following subsections.

2.3.1 Non-fragmented files

Non-fragmented or contiguous file is a common structure found in a forensic data set. Files in the dataset are in a consecutive order, even though one file uses two or more clusters to keep the data, but all clusters belonged to the file are contiguous as shown in Figure 2.2. File1 use two clusters but both clusters are adjacent to each other. It is the same with File3 where all three clusters are contiguous.

Figure 2.2: Contiguous files.

In file carving, most of the earliest forensic tools have the capability to carve contiguous files. According to Pal & Memon (2009), the first generation of file carvers used “magic number” which is byte sequences to be matched with the file metadata.

This is to identify and recover certain types of files (Li *et al.*, 2005). The "magic number" that has been used in this first generation is bytes that describe header or footer of the file (Sportiello & Zanero, 2012). Pal & Memon (2009) defined header as starting bytes of a file while footer as ending bytes of the files. Garfikel (2007) suggested alternatives such as header/maximum size carving for many file formats that does not restrict additional data appended to the end of a valid file, and header/embedded length carving for those files that have distinctive headers but no distinctive flag for the end. Besides that, Garfinkel (2007) also suggested file trimming for carving contiguous file. Trimming is the process of removing any data after the end of the object which is not part of the original file. However, Pal & Memon (2009) discussed the limitation of these first carvers. Some of the file types may not have footer but contains file size information such as BMP file. This makes the carver that uses header footer information fail to carve this kind of file. There is also issue of these first carvers simply extract data between the header and footer without certain that the data belong to the file. This is when fragmentation exists; part of file carved may contain parts or complete part of other file.

2.3.2 Fragmented files

Although there are many file carving tools but only few concentrate on carving fragmented files. Carving is relatively easy when all files are contiguous and using a technique identifying the header and footer but it is not easy dealing with fragmented files (Sportiello & Zanero, 2011). According to Garfinkel (2007) although the number of fragmented files are relatively small, but he discovered that the files that are of interest to forensics investigations are mostly fragmented. Furthermore, with huge increasing in digital data storage, criminals find new ways to hide their dirty work such as purposely fragmenting a digital image to make it hard to be recreated by the law enforcement (Xu & Dong, 2009).

Garfinkel (2007), Beek (2011) and Sencar & Memon (2009) mentioned that common modern operating systems always try to avoid fragmentation during writing process. However, fragmentation is unavoidable due to:

- unavailable contiguous region of sectors on the media that is large enough to accommodate the complete file. This happened when a drive has a long time

usage, almost full in capacity, and contains many files that require adding and deletion in roughly random order.

- data appending to existing files where in most cases, there may not be enough unallocated sectors at the end of the file to hold the new data.
- certain file systems like UNIX that do not support writing file in a contiguous way.

A fragmented file is a condition where a file is split into two or multiple parts which can be in different locations in a dataset (Kloet, 2007). In this case, bi-fragmented files will be an appealing target for automated carving because these files can be carved using uncomplicated algorithms compared to multiple fragmented files (Garfinkel, 2007). Figure 2.3 shows a condition where fragmentation occurs. File 1 is fragmented with File 2 where the second and third parts of File1 are after File 2.

Figure 2.3: Fragmented files

Pal *et al.* (2008) explained that fragmented files happened when a file is stored in non-contiguous clusters. When this happen, recovering files using traditional file carving technique will fail and if earlier automate file carving tool were utilized, incorrect files will be carved. Pal & Memon (2009) gave basic definitions concerning fragmentation as in Table 2.6.

Table 2.6: Term Definition

Term	Definition
CLUSTER	This is the size of the smallest data unit that can be written to disk and b_y will denote the cluster numbered *y* in the access order.
HEADER	This is a cluster that contains the starting point of a file.
FOOTER	This is a cluster that contains the ending data of a file
FRAGMENT	A fragment is considered to be one or more clusters of a file that are not sequentially connected to other clusters of the same file. Fragmented files are considered to have two or more fragments. Each fragment of a file is assumed to be separated from each other by unknown number of clusters.
BASE-FRAGMENT	The starting fragment of a file that contains the header as its first cluster.
FRAGMENTATION POINT	This is the last cluster belonging to a file before fragmentation occurs. A file may have multiple fragmentation point if it has multiple fragments.
FRAGMENTATION AREA	A set of consecutive clusters b_y, b_{y+1}, b_{y+2}, b_{y+3}… containing the fragmentation point.

Kloet (2007) categorized file fragmentation into two categories which are:

1. files with linear fragmentation where files are fragmented but all fragments present in a dataset in their original order.
2. files with non-linear fragmentation where files are fragmented but some of the fragments present in a different order not as in the original file.

In both categories, the crucial task is to identify fragmentation point which is explained in the next subsection.

2.3.2.1 Fragmentation point detection

Fragmentation point only exists when a file is fragmented into more than two parts. There are three approaches to detect fragmentation point that has been discussed in Pal *et al.* (2008) which are syntactical tests, statistical tests and basic sequential validation. Syntactical tests are used when the fragmentation point is detected by validating the belonging of a block to a file through one of following methods:

- Using keywords and signatures to identify different file types
- Content analysis to identify incorrect block.

This method can confirm that the validated block does not belong to any certain file. However, it is not for certain that the previous block belongs or does not belong to a particular file. For statistical tests, the statistic of each block is compared to a model of each file type to identify the block. Cohen (2007b) used the mapping function to map between the bytes contained in the file to the bytes within the image itself. In this case, carving process is the process to estimate the mapping function. Statistical tests also face problems in detecting the actual fragmentation point and even worse, using this technique, blocks can be falsely identified as belonging to other file types.

The third technique is basic sequential validation. This technique is used to identify fragmentation point by validating block sequentially from the header through the blocks until the validator stops with an error. Using this technique, the last correctly validated block is marked to be the fragmentation point. However, this technique can result in incorrect recovery of a file because it can wrongly validate random blocks of data.

REFERENCES

Alice Aartime's Photostream. Retrieved on October, 2011, from: http://www.flickr.com/photos/aartime/

Alex deSonnaville's Photostream. Retrieved on October, 2011, from: http://www.flickr.com/photos/desonnaville/

Alvarez, P. (2004). Using Extended File Information (*EXIF*) File Headers in Digital Evidence Analysis. *International Journal of Digital Evidence, 2(3),* pp. 1-4.

Antonio Sepulveda's Photostream. Retrieved on October, 2011, from: http://www.flickr.com/photos/58035092@N03/

Baguelin, F., Jacob, S., Malinge, C., Mouerier, J and Percot, F. (2010). Digital forensics framework. Retrieved on September,, 2010, from: http://www.digital-forensic.org/

Beek, C. (2011). *McAfee white paper on the introduction to file carving* [White paper]. Retrieved on October 2012, from: http://www.mcafee.com/au/resources/white-papers/foundstone/wp-intro-to-file-carving.pdf

Browne, P. S. (1972). Computer Security: A Survey. *ACM Sigmis Database, 4(3),* pp. 1-12.

Bettelli. S. (2006). The structure of JPEG of JPEG pictures. Retrieved on October 2012 from: http://cpan.uwinnipeg.ca/htdocs/Image-MetaData-JPEG/Structures.html

Carrier, B., Eoghan, C. and Wietse, V. *File carving challenge*. Retrieved on May, 2010, from: http://www.dfrws.org/2006/challenge/

Chai, I. & White, J. D. (2006). *Structuring Data and Building Algorithms: An ANSI Based Approach*. Singapore: McGraw Hill.

Cohen, K. (2007a). Digital still camera forensics. Small Scale Digital Device Forensics Journal, 1(1), pp. 1-8.

Cohen, M.I. (2007b). Advanced Carving Techniques. *Digital Investigation, 4(1-4),* pp. 119-128.

Commonwealth of Virginia Joint Commission on Technology and Science CVJCTS). Regional computer forensic laboratory (rcfl) national program office (npo),

http://jcots.state.va.us/2005%20Content/pdf/FBI-RCFL.pdf; September 8 2004.

Courrejou, T. and Garfinkel, S. L. (2011). *A comparative analysis of file carving software* (Report No. NPS-CS-11-006).

Digital Forensic Research Workshop. (2006). *DFRWS 2006 Forensics Challenge Details*. Retrieved on August, 2013 from: http://www.dfrws.org/2006/challenge/

Digital Forensic Research Workshop. (2007). *DFRWS 2006 Forensics Challenge Details*. Retrieved on August, 2013 from: http://www.dfrws.org/2007/challenge/

Exif.org. *Exif File's Samples*. Retrieved on October 2011 from: http://www.*Exif*.org/samples/fujifilm-mx1700.jpg

Fan, Z and de Queiroz, R. L. (2003). Identification of Bitmap Compression History: JPEG Detection and Quantizer Estimation. *IEEE Transactions on Image Processing, 12(2)*, pp. 230-235.

Garfinkel, S. L. (2007). Carving Contiguous and Fragmented Files with Fast Object Validation. *Digital Investigation, 4(1),* pp. S2-S12.

Garfinkel, S. L. Farrel, P., Roussev, V and Dinolt, G. (2009). Bringing Science to Digital Forensics with Standardized Forensic Corpora. *Digital Investigation, 6,* pp. S2-S11.

Garfinkel, S. L. (2010). Digital Forensics Research: The Next 10 Years. *Digital Investigation, 7(1),* pp. S64-S73.

Garfinkel, S. L. (2012). *File Carving*. Retrieved on October, 2011, from http://www.forensicswiki.org/wiki/File_Carving

Grenier, C. Data carving log. Retrieved on October, 2011, from http://www.cgsecurity.org/wiki/PhotoRec

Guo, H. and Xu, M. (2011). A method for recovering jpeg files based on thumbnail. *Proceeding of the International Conference Automation and Systems Engineering (CASE) 2011, (1-4),* pp. 2011.

Hamilton, E. (1992). JPEG File Interchange Format Version 1.02. Retrieved on August, 2013 from: http://www.w3.org/Graphics/JPEG/jfif3.pdf

Jemigan, R.P. and Quinn, S. D. *Two-pass Defragmentation of Compressed Hard Disk Data with a Single Data Rewrite*. U.S Patent 5574907. 1996.

Karresand, M. and Shahmehri, N. (2008). *Proceeding of the 2008 European Conference on Computer Network Defense*. pp. 28-32.

Kloet, S. J. J. *Measuring and Improving the Quality of File Carving Methods*. Master's Thesis. Eindhoven University of Technology; 2007.

Kornblum, J. D. (2008). Using JPEG Quantization Tables to Identify Imagery Processed by Software. *Digital Investigation, 5,* pp. S21-S25.

Li, W. J., Wang, K., Stolfo, S. J. and Herzog, B. (2005). Fileprints: Identifying File Types by n-gram Analysis. *Proceeding of the 2005 IEEE Workshop on Information Assurance and Security*. West Point, New York. pp. 64-71.

Li, Q., Sahin, B., Chang, E. C. and Thing, V. L. L. (2011). *Proceeding of the International Conference on Multimedia Computing and Systems.* pp. 1-6.

Linux manual. mkfs.xfs(8) : Linux man page. Retrieved on August, 2012 from http://linux.die.net/man/8/mkfs.xfs

Mitrakas, A. (2006). Law, Cyber Crime and Digital Forensics: Trailing Digital Suspects. in Kanellis, P., Kiountouzis, E., Kolokotronis, N. & Martakos, D. (Ed.). Digital Crime and Forensic Science in Cyberspace. Hershey: Idea Group Inc. pp. 267-290.

Merola, A. (2008). Data Carving Concepts [White paper]. Retrieved on August, 2013, from SANS Institute: http://www.sans.org/reading_room/whitepapers/forensics/data-carving-concepts

Metz, J. and Mora, R. J. (2006). Analysis of 2006 DFRWS Forensic Carving Challenge. Retrieved on August, 2013 from http://sandbox.dfrws.org/2006/mora/dfrws2006.pdf

Mikus, N. *An Analysis of Disc Carving Techniques.* Master's Thesis. Naval Postgraduate School; 2005.

Minami, S. and Zakhor, A. (1995). An Optimization Approach for Removing Blocking Effects in Transform Coding. *IEEE Transactions on Circuit and Systems for Video Technology, 5 (2),* pp. 74-82.

Memon, N. and Pal, A. (2006). Automated reassembly of the file fragmented images using greedy algorithms. *Journal IEEE Trans. Image Processing*, 15(2), pp. 385-393.

Mohamad, K. M. and Mat Deris, M. (2009a). Fragmentation Point Detection of JPEG Images at DHT Using Validator. *Proceeding of the 2009 FGIT*. pp. 173-180.

Mohamad, K. M. and Mat Deris, M. (2009b). Single-byte-marker for Detecting JPEG JFIF Header Using FORIMAGE-JPEG. *Proceeding of the 2009 Fifth International Joint Conference on INC, IMS and IDC.* pp. 1693-1698.

Mohamad, K. M. and Mat Deris, M. (2009c). Visualization of JPEG Metadata. In: H. R., Zaman, P., Robinson, M., Petrou, P., Oliver, H., Schroder &T. K., Shih. *Visual Informatics: Bridging research & Practises*. Springer Heidelberg. pp. 543-560.

Mohamad, K. M., Herawan, T. and Mat Deris, M. (2010a). Dual-byte-marker Algorithm for Detecting JFIF Header. In: Bandyopadyay, S. M., Adi, W., Kim, T. & Xiao, Y. (Ed.) *Information Security and Assurance*. Springer Heidelberg. pp. 17-26.

Mohamad, K. M., Patel, A., Herawan, T., Mat Deris, M. (2010b). myKarve: Jpeg image and thumbnail carver. *Journal of Digital Forensic Practice. 3*, pp.74-97.

Mohamad, K. M., Patel, A., Herawan, T., Mat Deris, M. (2011). Carving JPEG Images and Thumbnails Using Image Pattern Matching. *Proceeding of the 2011 IEEE Symposium on Computers & Informatics*. pp. 78-83.

Ng, S. W. Advances in Disk Technology: Performances Issues. (1998) *Computer*, *31*, pp. 75-81.

Pal, A., Shanmugasundram, K. and Memon, N. (2003). Automated Reassembly of Fragmented Images. *Proceeding of the 2003 Acoustics Speech and Signal Processing (ICASSP)*.

Pal, A., Sencar, H.T. and Memon, N. (2008). Detecting File Fragmentation Point Using Sequential Hypothesis Testing. *Digital Investigation, 5*, pp. S2-S13.

Pal, A. and Memon, N. (2009). The evolution of file carving. *IEEE Signal Processing Magazine*, 26(2). pp. 59-71. Retrieved October, 2011, from http://digital-assembly.com/technology/research/pubs/ieee-spm-2009.pdf.

Palmer, G.W. (2001). DFRWS Technical Report: A Road Map for Digital Forensic Research. Retrieved on August, 2013 from http://www.dfrws.org/2001/dfrws-rm-final.pdf

Povar, D. and Bhadran, V. K. (2011). Forensics Data Carving. *Digital Forensics and Cyber Crime, 53*, pp. 137-148.

Richard III, G. G. and Roussev, V. (2005). Scalpel: A Frugal, High Performance File Carver. *Proceeding of the 2005 Digital Forensics Research Workshop*. New Orleans.

Richard III, G. G., Roussev, V. and Marziale, L. (2007). In-place File Carving. *Research Advances in Digital Forensics III*, Springer.

Sencar, H. T. and Memon, N. (2009). Identification and Recovery of JPEG Files with Missing Fragments. *Digital Investigation*, 6, pp. S88-S98.

Shahbahrami, A., Bahrampour, R., Rostami, M. S., Mobarhan, M. A. (2011). Evaluation of Huffman and Arithmetic Algorithms for Multimedia Compression Standards. *International Journal of Computer Science, Engineering and Applications (IJCSEA), 1(4).*

Shannon, C. E. (2001). A mathematical theory of communication. *Bell System Technical Journal, 27,* pp. 379-423 and 623-656.

Shanmugasundram, K. and Memon, N. (2002). Automatic Reassembly of Document Fragments via Data Compression. *Proceeding of the 2002 Second Digital Forensics Research Workshop*. Syracuse.

Sitaraman, S. (2006). Computer and Network Forensics. in Kanellis, P., Kiountouzis, E., Kolokotronis, N. & Martakos, D. (Ed.). Digital Crime and Forensic Science in Cyberspace. Hershey: Idea Group Inc. pp. 55-74.

Sportiello, L. and Zanero, S. (2011). File Block Classification by Support Vector Machine. *Proceeding of Sixth International Conference on Availability, Reliability and Security (ARES)*.Vienna. pp. 307-312.

Sportiello, L., and Zanero, S. (2012).Context-Based File Block Classification. in G. Peterson, and S. Shenoi (Eds.). *Advances in Digital Forensics VIII*, Springer Berlin Heidelberg. pp. 67-82.

Stamm, M. C., Tjoa, S. K., Lin, W. S. and Liu, K. J. R. (2010). *Proceeding of the IEEE Int. Conf. Acoustics, Speech, and Signal Processing*. Dallas. pp. 1694-1697

Standard of Japan Electronics and Information Technology Industries Association (JEITA). *Exchangeable Image File Format for Digital Still Cameras Exif Version 2.1*. JEITA CP-3451. 2002.

Tesic, J. (2005). Metadata Practices for Consumer Photos. *IEEE Multimedia, 12 (3),* pp.86-92.

The International Telegraph and Telephone Consultative Committee (CCITT). *Information Technology Digital Compression and Coding of Continuous Tone Still Image Requirements and Guideline. T.81*. 1992.

Veenman, C. J. (2007). Statistical Disk Cluster Classification for File Carving. *The Third International Symposium on Information Assurance and Security*. Manchester. pp. 393-398.

Vinoth Chandar's Photostream. Retrieved on October, 2011, from: http://www.flickr.com/photos/vinothchandar/

Viraktamath, S. V. and Attimarad, G. V. (2011). Impact of Quantization Matrix on the Performance of JPEG. *International Journal of Future Generation Communication and Networking, 4 (3),* pp. 107-118.

Wallace, G. K. (1991). The JPEG still picture compression standard. *Communications of the ACM - Special issue on digital multimedia systems CACM, 34(4),* pp. 30-44.

Winzip computing. (2008). JPEG Compression. *JPEG Compression.doc Version 1.0.* pp. 1-20. Retrieved on October 2012 from: http://www.winzip.com/wz_jpg_comp.pdf

Wood, C. C., Banks, W. W. Guarro, S. B. Garcia, A. A., Hampel, V. E. and Sartorio, H. P. *Computer Security: A Comprehensive Control Checklist.* Wiley-Interscience, New York, NY, USA. 1987.

Xu, M. and Dong Shule. (2009). Reassembling the Fragmented JPEG Images Based on Sequential Pixel Prediction. *Proceeding of Computer Network and Multimedia Technology, 2009. CNMT 2009.* Wuhan. pp. 1-6.

Ying, H. M. and Thing, V. L. L. (2011). A Novel Inequality-based Fragmented File Carving Technique. *E- Forensics*, Springer. pp. 28-39.

Yu, Y, Lu Z., Ling, H., Zou, F. (2006). No-reference perceptual quality assessment of JPEG images using general regression neural network. *Advances in Neural Networks,* Springer-verlag berlin, pp. 638-645.

Zha, X. and Sahni, S. (2010). Fast In-place File Carving for Digital Forensics. *E-Forensics,* Springer. pp. 141-158.